Assignment 6
100 pts

There are two parts to this assignment. For the first part, you will carry out an SQL injection attack on a Chattr server. For the second, you will fix a number of security weaknesses in the Chattr application you developed in the second assignment. Your solution is due on December 4, 2014 no later than 10 P.M. This is an individual assignment. You must work alone. You may not discuss your solution with anyone until 7 days after the assignment deadline. See Section 4 for additional information on submitting your solution.

1 Part 1: SQL Injection

For this part of the assignment you will insert data into a database using an SQL injection attack. We are running a Chattr server at:

http://ian.ucsd.edu:8080/index.php

Joey Pardella has created an account on this Chattr server. You can view his posts at:


Your goal is to insert the message “XXXXXXXX is l33t” on Joey’s message board, where XXXXXXXX is your PGP key ID (8 hexadecimal digits). The post must obviously appear to be from Joey. For example, the TA PGP key ID is 7864D1BD, and a successful post would look like this:

<table>
<thead>
<tr>
<th>When</th>
<th>Who</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014-11-25 14:06:13.031256</td>
<td>pardellaj</td>
<td>7864D1BD is l33t</td>
</tr>
</tbody>
</table>

You may interact with the Chattr server in any way you wish. The assignment can be completed using SQL injection. You do not have access to the code or the schema, so it may take you several attempts to figure out how to do this.

2 Part 2: Securing Chattr

For this part of the assignment, you must fix any security vulnerabilities we have identified in your Chattr application from Assignment 2. Use the VM from the second assignment to develop your solution.

2.1 The Report

You will receive a report on the vulnerabilities we found in your Assignment 2 submission. Use this report as a guide to determine which problems you need to fix. You will be graded on the same set of vulnerabilities, although the tests may be slightly different.
2.2 Password Storage

Your application must store passwords salted. This means that if you stored the password in the clear or stored the password hashed using a fixed hash function (e.g. MD5), you will need to fix this part of the application.

2.3 Vulnerabilities

Your application must also protect against (a) SQL injection attacks, (b) cross-site scripting attacks, and (c) cross-site request forgery attacks. In other words, the attack must not be able to violate the Chattr security policy using these mechanisms. (The security policy was developed in Assignment 3.) The report you will receive will indicate which vulnerabilities we found.

3 Grading

The first part of the assignment is worth 30 points. You will receive 30 points if you successfully post your PGP key ID to Joey’s account, as described above.

The second part of the assignment is worth 70 points. We will use the same VM as in Assignment 2 to test your solution. Your solution must pass all the functional tests of Assignment 2. Therefore, you should first make sure you have a working solution for Assignment 2. If you are not sure if your solution passes the functional tests, your TAs will be happy to test your solution and let you know. We expect everyone to pass these tests.

If your solution passes the functional tests, your grade for this part of the assignment will depend on how many of the attacks we test succeed against your application (the more attacks succeed, the lower the grade). The attacks will be variations of SQL injection, cross-site scripting attacks, and cross-site request forgery, as identified in your report.

4 Submitting the Solution

Your solution to the first part of this assignment is a successful post to Joey Pardella’s Chattr page. Your attack must be completed by the assignment deadline (10 P.M. PST on December 4, 2014).

Your solution to the second part costs of the same PHP files you submitted for Assignment 2 (index.php, login.php, view.php, post.php, logout.php) as well as the database schema (db.sql). You may submit additional PHP files if your solution relies on them.

The files must be submitted via email to cs127f1@ieng6.ucsd.edu by December 4, 2014, 10:00 P.M. Pacific time. The files must be in a gzip-compressed tar archive, signed with your PGP key and encrypted to the cs127f1@ieng6.ucsd.edu PGP key. The cs127f1@ieng6.ucsd.edu key has key fingerprint:

ED49 BC3B 8992 A1E0 D2DD 66DC A1EF 6BAE 7864 D1BD.

To create a gzip-compressed tar archive, copy the files you wish to submit to single directory, change into that directory, and issue the command:

```
tar -zcvf /path/to/archive/{YOUR EMAIL}-hw6.tgz *.php *.sql
```

This will create an archive in the directory /path/to/archive/ containing all the PHP and SQL files in the current working directory.

To sign your submission with GPG:

```
gpg --encrypt --sign --armor -r cs127f1@ieng6.ucsd.edu {YOUR EMAIL}-hw6.tgz
```
For example, if the TAs were submitting a solution, their submission would be named

cse127f1@ieng6.ucsd.edu-hw6.tgz.gpg

There will be a 5-point penalty if your solution is not correctly signed and encrypted as described above.

5 Change History

This is the initial revision of the assignment.